

Grid

Grid The Grid system is a comprehensive CSS-based layout framework that provides responsive, mobile-first grid classes for building flexible and adaptive layouts. Based on Bootstrap's flexbox grid system, it includes containers, rows, columns, and extensive flexbox utilities with breakpoint-specific behavior. How to use ■ The Grid system is implemented as CSS classes that can be applied directly to HTML elements. No JavaScript or Angular components are required. <div class = " container " > <div class = " row " > <div class = " col-md-6 " > Column 1 </div> <div class = " col-md-6 " > Column 2 </div> </div> </div> Basic Grid ■ Simple grid layout with equal-width columns. Responsive Grid ■ Grid layout that adapts to different screen sizes using breakpoint-specific classes. Flexbox Utilities ■ Advanced flexbox utilities for controlling layout direction, alignment, and spacing. Alignment ■ Flexbox alignment utilities for controlling how items are positioned within containers. Ordering ■ CSS order utilities for controlling the visual order of flex items. Features ■ Container System ■ Fixed Container : .container with responsive max-widths Fluid Container : .container-fluid for full-width layouts Responsive Breakpoints : Automatic width adjustments at different screen sizes Consistent Padding : 15px horizontal padding on all containers Row System ■ Flexbox Rows : .row creates flex containers with wrap enabled Negative Margins : Automatic gutter compensation with -15px margins No Gutters : .no-gutters class removes spacing between columns Responsive Behavior : Consistent behavior across all breakpoints Column System ■ 12-Column Grid : Standard 12-column layout system Responsive Classes : Breakpoint-specific column classes (sm, md, lg, xl) Auto Columns : .col and .col-auto for flexible sizing Fixed Widths : Classes from .col-1 to .col-12 for precise control Breakpoint System ■ Extra Small (xs) : Default, no prefix (0px and up) Small (sm) : 576px and up Medium (md) : 768px and up Large (lg) : 992px and up Extra Large (xl) : 1200px and up Flexbox Utilities ■ Direction : Row, column, and reverse variants Wrap : Wrap, nowrap, and wrap-reverse options Justify Content : Start, end, center, between, around Align Items : Start, end, center, baseline, stretch Align Content : Start, end, center, between, around, stretch Align Self : Auto, start, end, center, baseline, stretch Positioning Utilities ■ Offsets : .offset-* classes for column positioning Push/Pull : .push-* and .pull-* for reordering Order : .order-* classes for flex item ordering Responsive Variants : All utilities available at each breakpoint CSS Classes ■ Container Classes ■ Class Description .container Fixed-width container with responsive max-widths .container-fluid Full-width container spanning the entire viewport Row Classes ■ Class Description .row Flexbox row container with wrap enabled .no-gutters Removes gutters from row and its columns Column Classes ■ Basic Columns ■ Class Width Description .col Auto Equal-width column that grows to fill space .col-auto Auto Column sized to content .col-1 8.333% 1/12 width column .col-2 16.667% 2/12 width column .col-3 25% 3/12 width column .col-4 33.333% 4/12 width column .col-5 41.667% 5/12 width column .col-6 50% 6/12 width column .col-7 58.333% 7/12 width column .col-8 66.667% 8/12 width column .col-9 75% 9/12 width

column .col-10 83.333% 10/12 width column .col-11 91.667% 11/12 width column .col-12 100% Full-width column

Responsive Columns

Class	Breakpoint Description
.col-sm-*	≥576px Small screen columns
.col-md-*	≥768px Medium screen columns
.col-lg-*	≥992px Large screen columns
.col-xl-*	≥1200px Extra large screen columns

Flexbox Direction Classes

Class	Description
.flex-row	Horizontal direction (default)
.flex-column	Vertical direction
.flex-row-reverse	Horizontal direction, reversed
.flex-column-reverse	Vertical direction, reversed

Flexbox Wrap Classes

Class	Description
.flex-wrap	Allow items to wrap to next line
.flex-nowrap	Prevent wrapping (default)
.flex-wrap-reverse	Wrap items in reverse order

Justify Content Classes

Class	Description
.justify-content-start	Align items to start
.justify-content-end	Align items to end
.justify-content-center	Center items
.justify-content-between	Space items evenly
.justify-content-around	Space items with equal margins

Align Items Classes

Class	Description
.align-items-start	Align items to start
.align-items-end	Align items to end
.align-items-center	Center items
.align-items-baseline	Align items to baseline
.align-items-stretch	Stretch items (default)

Align Content Classes

Class	Description
.align-content-start	Align content to start
.align-content-end	Align content to end
.align-content-center	Center content
.align-content-between	Space content evenly
.align-content-around	Space content with equal margins
.align-content-stretch	Stretch content (default)

Align Self Classes

Class	Description
.align-self-auto	Auto alignment (default)
.align-self-start	Align self to start
.align-self-end	Align self to end
.align-self-center	Center self
.align-self-baseline	Align self to baseline
.align-self-stretch	Stretch self

Order Classes

Class	Description
.order-first	Order: -1
.order-last	Order: 13
.order-1	Order: 1
.order-2	Order: 2
.order-3	Order: 3

Offset Classes

Class	Margin Left Description
.offset-1	8.333% Offset by 1 column
.offset-2	16.667% Offset by 2 columns
.offset-3	25% Offset by 3 columns
.offset-4	33.333% Offset by 4 columns
.offset-5	41.667% Offset by 5 columns
.offset-6	50% Offset by 6 columns
.offset-7	58.333% Offset by 7 columns
.offset-8	66.667% Offset by 8 columns
.offset-9	75% Offset by 9 columns
.offset-10	83.333% Offset by 10 columns
.offset-11	91.667% Offset by 11 columns

Push/Pull Classes

Class	Description
.push-*	Push columns to the right
.pull-*	Pull columns to the left

CSS Custom Properties

The Grid system uses standard CSS properties and doesn't rely on CSS custom properties. However, it's designed to work seamlessly with CSS custom properties for theming:

Property	Description	Default
--grid-gutter-width	Width of grid gutters	30px
--grid-columns	Number of grid columns	12
--container-max-widths	Container max-widths	540px, 720px, 960px, 1140px
--grid-breakpoints	Breakpoint values	576px, 768px, 992px, 1200px

Best Practices

- Layout Structure**
 - Container First** : Always start with a container (.container or .container-fluid)
 - Row Wrapper** : Wrap columns in a .row element
 - Column Classes** : Use appropriate column classes for desired widths
- Responsive Design** : Use breakpoint-specific classes for mobile-first design
- Responsive Design**
 - Mobile First** : Design for mobile devices first, then enhance for larger screens
 - Breakpoint Strategy** : Use consistent breakpoints across your application
 - Progressive Enhancement** : Start with basic layout, add complexity for larger screens
 - Content Priority** : Ensure important content is visible on all screen sizes
- Performance**
 - CSS Efficiency** : Grid classes are optimized for performance
 - Minimal Nesting** : Avoid deeply nested grid structures when possible
 - Semantic HTML** : Use semantic HTML elements with grid classes

Browser Support : Grid system works in all modern browsers Accessibility ■ Logical Order : Ensure content flows logically regardless of visual order Screen Readers : Grid layout doesn't affect screen reader navigation Keyboard Navigation : Maintain logical tab order Focus Management : Ensure focus indicators are visible Content Organization ■ Consistent Spacing : Use consistent gutters and spacing Content Hierarchy : Use grid to establish clear visual hierarchy Flexible Content : Design content to work across different column widths Responsive Images : Ensure images scale appropriately within grid columns Accessibility Guidelines ■ Semantic Structure ■ HTML Semantics : Use semantic HTML elements with grid classes Document Outline : Ensure proper heading hierarchy within grid layouts Landmark Roles : Use appropriate ARIA landmarks for major layout sections Content Flow : Maintain logical content flow regardless of visual layout Screen Reader Support ■ Logical Order : Content should flow logically for screen readers Descriptive Labels : Use descriptive labels for interactive elements Skip Links : Provide skip links for navigation-heavy layouts Focus Indicators : Ensure focus indicators are visible and logical Keyboard Navigation ■ Tab Order : Maintain logical tab order through grid layouts Focus Management : Ensure focus moves logically through content Keyboard Shortcuts : Provide keyboard shortcuts for complex layouts Escape Mechanisms : Allow users to escape from complex layouts Visual Design ■ High Contrast : Ensure sufficient contrast ratios in grid layouts Color Independence : Don't rely solely on color for information Text Scaling : Support for text scaling and zoom Motion Sensitivity : Provide alternatives for users sensitive to motion Responsive Behavior ■ Mobile Adaptations ■ The Grid system automatically adapts to mobile screens: Stacked Layout : Columns stack vertically on small screens Full Width : Single columns take full width on mobile Touch Optimization : Layouts are optimized for touch interaction Performance : Optimized performance for mobile devices Breakpoint Behavior ■ Extra Small (0px+) : Single column layout, stacked content Small (576px+) : Multi-column layouts begin to appear Medium (768px+) : Tablet-optimized layouts Large (992px+) : Desktop layouts with full feature set Extra Large (1200px+) : Large desktop and wide screen layouts Content Considerations ■ Flexible Images : Images scale appropriately within grid columns Text Readability : Text remains readable at all column widths Touch Targets : Interactive elements maintain appropriate touch target sizes Loading Performance : Layouts load efficiently on all devices

```

<div class="grid-basic-demo">
  <h3>Basic Grid Layout</h3>

  <div class="demo-section">
    <h4>Equal Width Columns</h4>
    <p>
      Basic grid with equal-width columns using the <code>.col</code> class.
    </p>

    <div class="container">
      <div class="row">
        <div class="col">
          <div class="grid-item">
            <h5>Column 1</h5>
            <p>This column automatically sizes to fill available space.</p>
          </div>
        </div>
        <div class="col">
          <div class="grid-item">
            <h5>Column 2</h5>
            <p>This column also automatically sizes to fill available space.</p>
          </div>
        </div>
        <div class="col">
          <div class="grid-item">
            <h5>Column 3</h5>
            <p>All columns in this row have equal width.</p>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="demo-section">
    <h4>Fixed Width Columns</h4>
    <p>Grid with specific column widths using numbered classes.</p>

    <div class="container">
      <div class="row">
        <div class="col-8">
          <div class="grid-item primary">
            <h5>Main Content (8/12)</h5>
            <p>This column takes up 8 out of 12 grid columns (66.67% width).</p>
          </div>
        </div>
        <div class="col-4">
          <div class="grid-item secondary">
            <h5>Sidebar (4/12)</h5>
            <p>This column takes up 4 out of 12 grid columns (33.33% width).</p>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="demo-section">

```

<h4>Mixed Column Types</h4>

<p>Combining fixed-width and auto-width columns.</p>

```
<div class="container">
  <div class="row">
    <div class="col-6">
      <div class="grid-item">
        <h5>Fixed Width (6/12)</h5>
        <p>This column has a fixed width of 50%.</p>
      </div>
    </div>
    <div class="col">
      <div class="grid-item">
        <h5>Auto Width</h5>
        <p>This column automatically sizes to fill remaining space.</p>
      </div>
    </div>
    <div class="col-auto">
      <div class="grid-item compact">
        <h5>Auto Size</h5>
        <p>Fits content.</p>
      </div>
    </div>
  </div>
</div>
</div>
```

<div class="demo-section">

<h4>Container Types</h4>

<p>Different container types for different layout needs.</p>

```
<div class="container">
  <div class="row">
    <div class="col-12">
      <div class="grid-item">
        <h5>Fixed Container</h5>
        <p>
          This uses <code>.container</code> with responsive max-widths and
          centered content.
        </p>
      </div>
    </div>
  </div>
</div>
```

```
<div class="container-fluid">
  <div class="row">
    <div class="col-12">
      <div class="grid-item fluid">
        <h5>Fluid Container</h5>
        <p>
          This uses <code>.container-fluid</code> for full-width layout
          spanning the entire viewport.
        </p>
      </div>
    </div>
  </div>
</div>
```

```

    </div>
  </div>
</div>

<div class="demo-section">
  <h4>No Gutters</h4>
  <p>Grid without gutters using the <code>.no-gutters</code> class.</p>

  <div class="container">
    <div class="row no-gutters">
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>This column has no spacing between it and adjacent columns.</p>
        </div>
      </div>
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>Adjacent columns touch each other with no gap.</p>
        </div>
      </div>
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>Perfect for seamless layouts and full-width designs.</p>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="demo-info">
  <h4>Basic Grid Features</h4>
  <ul>
    <li>
      <strong>Container System:</strong> Fixed-width and fluid containers
    </li>
    <li>
      <strong>Row Wrapper:</strong> Flexbox rows with automatic gutter management
    </li>
    <li>
      <strong>Column Classes:</strong> Equal-width, fixed-width, and auto-width options
    </li>
    <li>
      <strong>No Gutters:</strong> Option to remove spacing between columns
    </li>
    <li>
      <strong>Responsive Foundation:</strong> Mobile-first approach with responsive breakpoints
    </li>
  </ul>
</div>
</div>

```

```
import { Component } from "@angular/core";

@Component({
  selector: "app-grid-basic",
  standalone: true,
  template: `
    <div class="grid-basic-demo">
      <h3>Basic Grid Layout</h3>

      <div class="demo-section">
        <h4>Equal Width Columns</h4>
        <p>
          Basic grid with equal-width columns using the <code>.col</code> class.
        </p>

        <div class="container">
          <div class="row">
            <div class="col">
              <div class="grid-item">
                <h5>Column 1</h5>
                <p>This column automatically sizes to fill available space.</p>
              </div>
            </div>
            <div class="col">
              <div class="grid-item">
                <h5>Column 2</h5>
                <p>
                  This column also automatically sizes to fill available space.
                </p>
              </div>
            </div>
            <div class="col">
              <div class="grid-item">
                <h5>Column 3</h5>
                <p>All columns in this row have equal width.</p>
              </div>
            </div>
          </div>
        </div>

        <div class="demo-section">
          <h4>Fixed Width Columns</h4>
          <p>Grid with specific column widths using numbered classes.</p>

          <div class="container">
            <div class="row">
              <div class="col-8">
                <div class="grid-item primary">
                  <h5>Main Content (8/12)</h5>
                  <p>
                    This column takes up 8 out of 12 grid columns (66.67% width).
                  </p>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  `
})
```

```

    </div>
    <div class="col-4">
      <div class="grid-item secondary">
        <h5>Sidebar (4/12)</h5>
        <p>
          This column takes up 4 out of 12 grid columns (33.33% width).
        </p>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>

<div class="demo-section">
  <h4>Mixed Column Types</h4>
  <p>Combining fixed-width and auto-width columns.</p>

  <div class="container">
    <div class="row">
      <div class="col-6">
        <div class="grid-item">
          <h5>Fixed Width (6/12)</h5>
          <p>This column has a fixed width of 50%.</p>
        </div>
      </div>
      <div class="col">
        <div class="grid-item">
          <h5>Auto Width</h5>
          <p>This column automatically sizes to fill remaining space.</p>
        </div>
      </div>
      <div class="col-auto">
        <div class="grid-item compact">
          <h5>Auto Size</h5>
          <p>Fits content.</p>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="demo-section">
  <h4>Container Types</h4>
  <p>Different container types for different layout needs.</p>

  <div class="container">
    <div class="row">
      <div class="col-12">
        <div class="grid-item">
          <h5>Fixed Container</h5>
          <p>
            This uses <code>.container</code> with responsive max-widths
            and centered content.
          </p>
        </div>
      </div>
    </div>
  </div>
</div>

```



```

    </div>
</div>

<div class="container-fluid">
  <div class="row">
    <div class="col-12">
      <div class="grid-item fluid">
        <h5>Fluid Container</h5>
        <p>
          This uses <code>.container-fluid</code> for full-width layout
          spanning the entire viewport.
        </p>
      </div>
    </div>
  </div>
</div>

<div class="demo-section">
  <h4>No Gutters</h4>
  <p>Grid without gutters using the <code>.no-gutters</code> class.</p>

  <div class="container">
    <div class="row no-gutters">
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>
            This column has no spacing between it and adjacent columns.
          </p>
        </div>
      </div>
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>Adjacent columns touch each other with no gap.</p>
        </div>
      </div>
      <div class="col-4">
        <div class="grid-item no-gutter">
          <h5>No Gutter</h5>
          <p>Perfect for seamless layouts and full-width designs.</p>
        </div>
      </div>
    </div>
  </div>

  <div class="demo-info">
    <h4>Basic Grid Features</h4>
    <ul>
      <li>
        <strong>Container System:</strong> Fixed-width and fluid containers
      </li>
      <li>
        <strong>Row Wrapper:</strong> Flexbox rows with automatic gutter
      </li>
    </ul>
  </div>

```

```

        management
    </li>
    <li>
        <strong>Column Classes:</strong> Equal-width, fixed-width, and
        auto-width options
    </li>
    <li>
        <strong>No Gutters:</strong> Option to remove spacing between
        columns
    </li>
    <li>
        <strong>Responsive Foundation:</strong> Mobile-first approach with
        responsive breakpoints
    </li>
</ul>
</div>
</div>
,
styles: [
    ,
    .grid-basic-demo {
        padding: 20px;
        max-width: 1200px;
        margin: 0 auto;
    }

    .grid-basic-demo h3 {
        margin: 0 0 20px 0;
        color: #333;
        font-size: 24px;
        font-weight: 600;
    }

    .demo-section {
        margin-bottom: 40px;
        padding: 20px;
        border: 1px solid #e5e7eb;
        border-radius: 8px;
        background: #f9fafb;
    }

    .demo-section h4 {
        margin: 0 0 12px 0;
        color: #333;
        font-size: 18px;
        font-weight: 600;
    }

    .demo-section p {
        margin: 0 0 20px 0;
        color: #555;
        line-height: 1.5;
    }

    .demo-section code {
        background: #f1f3f4;

```

```
padding: 2px 6px;
border-radius: 4px;
font-family: monospace;
font-size: 14px;
}

.grid-item {
  background: #fff;
  border: 1px solid #e5e7eb;
  border-radius: 6px;
  padding: 20px;
  height: 100%;
  min-height: 120px;
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.grid-item h5 {
  margin: 0 0 12px 0;
  color: #333;
  font-size: 16px;
  font-weight: 600;
}

.grid-item p {
  margin: 0;
  color: #666;
  line-height: 1.4;
}

.grid-item.primary {
  background: #e3f2fd;
  border-color: #2196f3;
}

.grid-item.secondary {
  background: #f3e5f5;
  border-color: #9c27b0;
}

.grid-item.compact {
  min-height: 80px;
  padding: 15px;
}

.grid-item.fluid {
  background: #e8f5e8;
  border-color: #4caf50;
}

.grid-item.no-gutter {
  border-radius: 0;
  border-left: none;
}
```

```

.grid-item.no-gutter:first-child {
  border-left: 1px solid #e5e7eb;
  border-top-left-radius: 6px;
  border-bottom-left-radius: 6px;
}

.grid-item.no-gutter:last-child {
  border-top-right-radius: 6px;
  border-bottom-right-radius: 6px;
}

.demo-info {
  margin-top: 40px;
  padding: 20px;
  background: #f8f9fa;
  border: 1px solid #dee2e6;
  border-radius: 8px;
}

.demo-info h4 {
  margin: 0 0 16px 0;
  color: #333;
  font-size: 18px;
  font-weight: 600;
}

.demo-info ul {
  margin: 0;
  padding-left: 20px;
  color: #555;
}

.demo-info li {
  margin-bottom: 8px;
  line-height: 1.4;
}

.demo-info strong {
  color: #333;
}
},
],
})
export class GridBasicComponent {
  constructor() {
    console.log("Grid Basic Component initialized");
  }
}

```

```

<div class="grid-responsive-demo">
  <h3>Responsive Grid Layout</h3>

  <div class="demo-section">
    <h4>Responsive Column Classes</h4>
    <p>
      Grid columns that adapt to different screen sizes using
      breakpoint-specific classes.
    </p>

    <div class="container">
      <div class="row">
        <div class="col-12 col-sm-6 col-md-4 col-lg-3">
          <div class="grid-item responsive">
            <h5>Responsive Column</h5>
            <p>
              <strong>Mobile:</strong> 12/12 (100%)<br />
              <strong>Small:</strong> 6/12 (50%)<br />
              <strong>Medium:</strong> 4/12 (33.33%)<br />
              <strong>Large:</strong> 3/12 (25%)
            </p>
          </div>
        </div>
      </div>
      <div class="col-12 col-sm-6 col-md-4 col-lg-3">
        <div class="grid-item responsive">
          <h5>Responsive Column</h5>
          <p>
            <strong>Mobile:</strong> 12/12 (100%)<br />
            <strong>Small:</strong> 6/12 (50%)<br />
            <strong>Medium:</strong> 4/12 (33.33%)<br />
            <strong>Large:</strong> 3/12 (25%)
          </p>
        </div>
      </div>
      <div class="col-12 col-sm-6 col-md-4 col-lg-3">
        <div class="grid-item responsive">
          <h5>Responsive Column</h5>
          <p>
            <strong>Mobile:</strong> 12/12 (100%)<br />
            <strong>Small:</strong> 6/12 (50%)<br />
            <strong>Medium:</strong> 4/12 (33.33%)<br />
            <strong>Large:</strong> 3/12 (25%)
          </p>
        </div>
      </div>
      <div class="col-12 col-sm-6 col-md-4 col-lg-3">
        <div class="grid-item responsive">
          <h5>Responsive Column</h5>
          <p>
            <strong>Mobile:</strong> 12/12 (100%)<br />
            <strong>Small:</strong> 6/12 (50%)<br />
            <strong>Medium:</strong> 4/12 (33.33%)<br />
            <strong>Large:</strong> 3/12 (25%)
          </p>
        </div>
      </div>
    </div>
  </div>

```

```

    </div>
  </div>
</div>

<div class="demo-section">
  <h4>Mobile-First Layout</h4>
  <p>
    Layout that starts with mobile design and progressively enhances for
    larger screens.
  </p>

  <div class="container">
    <div class="row">
      <div class="col-12 col-md-8">
        <div class="grid-item primary">
          <h5>Main Content</h5>
          <p>
            <strong>Mobile:</strong> Full width (12/12)<br />
            <strong>Medium+:</strong> 8/12 width
          </p>
          <p>
            This column takes full width on mobile devices and 8/12 width on
            medium screens and larger.
          </p>
        </div>
      </div>
      <div class="col-12 col-md-4">
        <div class="grid-item secondary">
          <h5>Sidebar</h5>
          <p>
            <strong>Mobile:</strong> Full width (12/12)<br />
            <strong>Medium+:</strong> 4/12 width
          </p>
          <p>
            This sidebar appears below the main content on mobile and beside
            it on larger screens.
          </p>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="demo-section">
  <h4>Complex Responsive Layout</h4>
  <p>
    Advanced responsive layout with different column configurations at each
    breakpoint.
  </p>

  <div class="container">
    <div class="row">
      <div class="col-12 col-sm-6 col-lg-4">
        <div class="grid-item">
          <h5>Feature 1</h5>
          <p>

```

```

        <strong>Mobile:</strong> 12/12 (100%)<br />
        <strong>Small:</strong> 6/12 (50%)<br />
        <strong>Large:</strong> 4/12 (33.33%)
    </p>
</div>
</div>
<div class="col-12 col-sm-6 col-lg-4">
    <div class="grid-item">
        <h5>Feature 2</h5>
        <p>
            <strong>Mobile:</strong> 12/12 (100%)<br />
            <strong>Small:</strong> 6/12 (50%)<br />
            <strong>Large:</strong> 4/12 (33.33%)
        </p>
    </div>
</div>
<div class="col-12 col-sm-12 col-lg-4">
    <div class="grid-item">
        <h5>Feature 3</h5>
        <p>
            <strong>Mobile:</strong> 12/12 (100%)<br />
            <strong>Small:</strong> 12/12 (100%)<br />
            <strong>Large:</strong> 4/12 (33.33%)
        </p>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="demo-section">
    <h4>Responsive Navigation</h4>
    <p>Navigation layout that adapts to different screen sizes.</p>

    <div class="container">
        <div class="row align-items-center">
            <div class="col-12 col-md-3">
                <div class="grid-item nav-brand">
                    <h5>Brand Logo</h5>
                    <p>Always full width on mobile, 3/12 on medium+</p>
                </div>
            </div>
            <div class="col-12 col-md-9">
                <div class="grid-item nav-menu">
                    <h5>Navigation Menu</h5>
                    <p>Full width on mobile, 9/12 on medium+</p>
                    <div class="nav-items">
                        <span class="nav-item">Home</span>
                        <span class="nav-item">About</span>
                        <span class="nav-item">Services</span>
                        <span class="nav-item">Contact</span>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>

```

</div>

<div class="demo-section">

<h4>Responsive Form Layout</h4>

<p>

Form fields that stack on mobile and align side-by-side on larger screens.

</p>

<div class="container">

<div class="row">

<div class="col-12 col-sm-6">

<div class="grid-item form-field">

<h5>First Name</h5>

<p>Full width on mobile, 6/12 on small+</p>

</div>

</div>

<div class="col-12 col-sm-6">

<div class="grid-item form-field">

<h5>Last Name</h5>

<p>Full width on mobile, 6/12 on small+</p>

</div>

</div>

<div class="col-12">

<div class="grid-item form-field">

<h5>Email Address</h5>

<p>Always full width across all screen sizes</p>

</div>

</div>

<div class="col-12 col-md-8">

<div class="grid-item form-field">

<h5>Message</h5>

<p>Full width on mobile, 8/12 on medium+</p>

</div>

</div>

<div class="col-12 col-md-4">

<div class="grid-item form-field">

<h5>Submit</h5>

<p>Full width on mobile, 4/12 on medium+</p>

</div>

</div>

</div>

</div>

</div>

<div class="demo-info">

<h4>Responsive Grid Features</h4>

Mobile-First: Design for mobile devices first, then enhance for larger screens

Breakpoint Classes: Use <code>.col-sm-*</code>, <code>.col-md-*</code>, <code>.col-lg-*</code>, <code>.col-xl-*</code>

- Progressive Enhancement:** Start with basic layout, add complexity for larger screens
- Content Priority:** Ensure important content is visible on all screen sizes
- Touch Optimization:** Layouts are optimized for touch interaction on mobile

Breakpoint Reference

- Extra Small (xs):** 0px and up (default, no prefix)
- Small (sm):** 576px and up
- Medium (md):** 768px and up
- Large (lg):** 992px and up
- Extra Large (xl):** 1200px and up

```
import { Component } from "@angular/core";
```

```
@Component({
  selector: "app-grid-responsive",
  standalone: true,
  template: `
    <div class="grid-responsive-demo">
      <h3>Responsive Grid Layout</h3>

      <div class="demo-section">
        <h4>Responsive Column Classes</h4>
        <p>
          Grid columns that adapt to different screen sizes using
          breakpoint-specific classes.
        </p>

        <div class="container">
          <div class="row">
            <div class="col-12 col-sm-6 col-md-4 col-lg-3">
              <div class="grid-item responsive">
                <h5>Responsive Column</h5>
                <p>
                  <strong>Mobile:</strong> 12/12 (100%)<br />
                  <strong>Small:</strong> 6/12 (50%)<br />
                  <strong>Medium:</strong> 4/12 (33.33%)<br />
                  <strong>Large:</strong> 3/12 (25%)
                </p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  `
})
```

```

    </div>
    <div class="col-12 col-sm-6 col-md-4 col-lg-3">
      <div class="grid-item responsive">
        <h5>Responsive Column</h5>
        <p>
          <strong>Mobile:</strong> 12/12 (100%)<br />
          <strong>Small:</strong> 6/12 (50%)<br />
          <strong>Medium:</strong> 4/12 (33.33%)<br />
          <strong>Large:</strong> 3/12 (25%)
        </p>
      </div>
    </div>
    <div class="col-12 col-sm-6 col-md-4 col-lg-3">
      <div class="grid-item responsive">
        <h5>Responsive Column</h5>
        <p>
          <strong>Mobile:</strong> 12/12 (100%)<br />
          <strong>Small:</strong> 6/12 (50%)<br />
          <strong>Medium:</strong> 4/12 (33.33%)<br />
          <strong>Large:</strong> 3/12 (25%)
        </p>
      </div>
    </div>
    <div class="col-12 col-sm-6 col-md-4 col-lg-3">
      <div class="grid-item responsive">
        <h5>Responsive Column</h5>
        <p>
          <strong>Mobile:</strong> 12/12 (100%)<br />
          <strong>Small:</strong> 6/12 (50%)<br />
          <strong>Medium:</strong> 4/12 (33.33%)<br />
          <strong>Large:</strong> 3/12 (25%)
        </p>
      </div>
    </div>
  </div>
</div>
<div class="demo-section">
  <h4>Mobile-First Layout</h4>
  <p>
    Layout that starts with mobile design and progressively enhances for
    larger screens.
  </p>
  <div class="container">
    <div class="row">
      <div class="col-12 col-md-8">
        <div class="grid-item primary">
          <h5>Main Content</h5>
          <p>
            <strong>Mobile:</strong> Full width (12/12)<br />
            <strong>Medium+:</strong> 8/12 width
          </p>
          <p>

```

```

        This column takes full width on mobile devices and 8/12 width
        on medium screens and larger.
    </p>
</div>
</div>
<div class="col-12 col-md-4">
    <div class="grid-item secondary">
        <h5>Sidebar</h5>
        <p>
            <strong>Mobile:</strong> Full width (12/12)<br />
            <strong>Medium+:</strong> 4/12 width
        </p>
        <p>
            This sidebar appears below the main content on mobile and
            beside it on larger screens.
        </p>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="demo-section">
    <h4>Complex Responsive Layout</h4>
    <p>
        Advanced responsive layout with different column configurations at
        each breakpoint.
    </p>

    <div class="container">
        <div class="row">
            <div class="col-12 col-sm-6 col-lg-4">
                <div class="grid-item">
                    <h5>Feature 1</h5>
                    <p>
                        <strong>Mobile:</strong> 12/12 (100%)<br />
                        <strong>Small:</strong> 6/12 (50%)<br />
                        <strong>Large:</strong> 4/12 (33.33%)
                    </p>
                </div>
            </div>
            <div class="col-12 col-sm-6 col-lg-4">
                <div class="grid-item">
                    <h5>Feature 2</h5>
                    <p>
                        <strong>Mobile:</strong> 12/12 (100%)<br />
                        <strong>Small:</strong> 6/12 (50%)<br />
                        <strong>Large:</strong> 4/12 (33.33%)
                    </p>
                </div>
            </div>
            <div class="col-12 col-sm-12 col-lg-4">
                <div class="grid-item">
                    <h5>Feature 3</h5>
                    <p>
                        <strong>Mobile:</strong> 12/12 (100%)<br />

```

```

        <strong>Small:</strong> 12/12 (100%)<br />
        <strong>Large:</strong> 4/12 (33.33%)
    </p>
</div>
</div>
</div>
</div>
</div>
</div>

<div class="demo-section">
    <h4>Responsive Navigation</h4>
    <p>Navigation layout that adapts to different screen sizes.</p>

    <div class="container">
        <div class="row align-items-center">
            <div class="col-12 col-md-3">
                <div class="grid-item nav-brand">
                    <h5>Brand Logo</h5>
                    <p>Always full width on mobile, 3/12 on medium+</p>
                </div>
            </div>
            <div class="col-12 col-md-9">
                <div class="grid-item nav-menu">
                    <h5>Navigation Menu</h5>
                    <p>Full width on mobile, 9/12 on medium+</p>
                    <div class="nav-items">
                        <span class="nav-item">Home</span>
                        <span class="nav-item">About</span>
                        <span class="nav-item">Services</span>
                        <span class="nav-item">Contact</span>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<div class="demo-section">
    <h4>Responsive Form Layout</h4>
    <p>
        Form fields that stack on mobile and align side-by-side on larger
        screens.
    </p>

    <div class="container">
        <div class="row">
            <div class="col-12 col-sm-6">
                <div class="grid-item form-field">
                    <h5>First Name</h5>
                    <p>Full width on mobile, 6/12 on small+</p>
                </div>
            </div>
            <div class="col-12 col-sm-6">
                <div class="grid-item form-field">
                    <h5>Last Name</h5>
                    <p>Full width on mobile, 6/12 on small+</p>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
  <div class="col-12">
    <div class="grid-item form-field">
      <h5>Email Address</h5>
      <p>Always full width across all screen sizes</p>
    </div>
  </div>
  <div class="col-12 col-md-8">
    <div class="grid-item form-field">
      <h5>Message</h5>
      <p>Full width on mobile, 8/12 on medium+</p>
    </div>
  </div>
  <div class="col-12 col-md-4">
    <div class="grid-item form-field">
      <h5>Submit</h5>
      <p>Full width on mobile, 4/12 on medium+</p>
    </div>
  </div>
</div>
</div>
</div>
</div>

<div class="demo-info">
  <h4>Responsive Grid Features</h4>
  <ul>
    <li>
      <strong>Mobile-First:</strong> Design for mobile devices first, then
      enhance for larger screens
    </li>
    <li>
      <strong>Breakpoint Classes:</strong> Use <code>.col-sm-*</code>,
      <code>.col-md-*</code>, <code>.col-lg-*</code>,
      <code>.col-xl-*</code>
    </li>
    <li>
      <strong>Progressive Enhancement:</strong> Start with basic layout,
      add complexity for larger screens
    </li>
    <li>
      <strong>Content Priority:</strong> Ensure important content is
      visible on all screen sizes
    </li>
    <li>
      <strong>Touch Optimization:</strong> Layouts are optimized for touch
      interaction on mobile
    </li>
  </ul>

  <h4>Breakpoint Reference</h4>
  <ul>
    <li>
      <strong>Extra Small (xs):</strong> 0px and up (default, no prefix)
    </li>
    <li><strong>Small (sm):</strong> 576px and up</li>
  </ul>

```

```

        <li><strong>Medium (md):</strong> 768px and up</li>
        <li><strong>Large (lg):</strong> 992px and up</li>
        <li><strong>Extra Large (xl):</strong> 1200px and up</li>
    </ul>
</div>
</div>
,
styles: [
    ,
    .grid-responsive-demo {
        padding: 20px;
        max-width: 1200px;
        margin: 0 auto;
    }

    .grid-responsive-demo h3 {
        margin: 0 0 20px 0;
        color: #333;
        font-size: 24px;
        font-weight: 600;
    }

    .demo-section {
        margin-bottom: 40px;
        padding: 20px;
        border: 1px solid #e5e7eb;
        border-radius: 8px;
        background: #f9fafb;
    }

    .demo-section h4 {
        margin: 0 0 12px 0;
        color: #333;
        font-size: 18px;
        font-weight: 600;
    }

    .demo-section p {
        margin: 0 0 20px 0;
        color: #555;
        line-height: 1.5;
    }

    .demo-section code {
        background: #f1f3f4;
        padding: 2px 6px;
        border-radius: 4px;
        font-family: monospace;
        font-size: 14px;
    }

    .grid-item {
        background: #fff;
        border: 1px solid #e5e7eb;
        border-radius: 6px;
        padding: 20px;
    }

```

```
    height: 100%;
    min-height: 120px;
    display: flex;
    flex-direction: column;
    justify-content: center;
}

.grid-item h5 {
    margin: 0 0 12px 0;
    color: #333;
    font-size: 16px;
    font-weight: 600;
}

.grid-item p {
    margin: 0;
    color: #666;
    line-height: 1.4;
}

.grid-item.responsive {
    background: #fff3cd;
    border-color: #ffc107;
}

.grid-item.primary {
    background: #e3f2fd;
    border-color: #2196f3;
}

.grid-item.secondary {
    background: #f3e5f5;
    border-color: #9c27b0;
}

.grid-item.nav-brand {
    background: #e8f5e8;
    border-color: #4caf50;
    min-height: 80px;
}

.grid-item.nav-menu {
    background: #fff8e1;
    border-color: #ff9800;
    min-height: 80px;
}

.nav-items {
    display: flex;
    gap: 15px;
    margin-top: 10px;
}

.nav-item {
    background: #ff9800;
    color: white;
```

```

        padding: 5px 10px;
        border-radius: 4px;
        font-size: 14px;
    }

    .grid-item.form-field {
        background: #f8f9fa;
        border-color: #6c757d;
        min-height: 100px;
    }

    .demo-info {
        margin-top: 40px;
        padding: 20px;
        background: #f8f9fa;
        border: 1px solid #dee2e6;
        border-radius: 8px;
    }

    .demo-info h4 {
        margin: 0 0 16px 0;
        color: #333;
        font-size: 18px;
        font-weight: 600;
    }

    .demo-info ul {
        margin: 0 0 20px 0;
        padding-left: 20px;
        color: #555;
    }

    .demo-info li {
        margin-bottom: 8px;
        line-height: 1.4;
    }

    .demo-info strong {
        color: #333;
    }
},
],
}))
export class GridResponsiveComponent {
    constructor() {
        console.log("Grid Responsive Component initialized");
    }
}

```



```

<div class="grid-flexbox-demo">
  <h3>Flexbox Utilities</h3>

  <div class="demo-section">
    <h4>Flex Direction</h4>
    <p>Control the direction of flex items using direction classes.</p>

    <div class="container">
      <div class="row">
        <div class="col-6">
          <div class="grid-item">
            <h5>Row Direction (Default)</h5>
            <p>Items flow horizontally from left to right.</p>
            <div class="flex-demo flex-row">
              <div class="flex-item">1</div>
              <div class="flex-item">2</div>
              <div class="flex-item">3</div>
            </div>
          </div>
        </div>
        <div class="col-6">
          <div class="grid-item">
            <h5>Column Direction</h5>
            <p>Items flow vertically from top to bottom.</p>
            <div class="flex-demo flex-column">
              <div class="flex-item">1</div>
              <div class="flex-item">2</div>
              <div class="flex-item">3</div>
            </div>
          </div>
        </div>
      </div>

      <div class="row">
        <div class="col-6">
          <div class="grid-item">
            <h5>Row Reverse</h5>
            <p>Items flow horizontally from right to left.</p>
            <div class="flex-demo flex-row-reverse">
              <div class="flex-item">1</div>
              <div class="flex-item">2</div>
              <div class="flex-item">3</div>
            </div>
          </div>
        </div>
        <div class="col-6">
          <div class="grid-item">
            <h5>Column Reverse</h5>
            <p>Items flow vertically from bottom to top.</p>
            <div class="flex-demo flex-column-reverse">
              <div class="flex-item">1</div>
              <div class="flex-item">2</div>
              <div class="flex-item">3</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    </div>
  </div>
</div>

<div class="demo-section">
  <h4>Flex Wrap</h4>
  <p>Control how flex items wrap when they exceed the container width.</p>

  <div class="container">
    <div class="row">
      <div class="col-6">
        <div class="grid-item">
          <h5>Wrap (Default)</h5>
          <p>Items wrap to the next line when needed.</p>
          <div class="flex-demo flex-wrap">
            <div class="flex-item wide">Item 1</div>
            <div class="flex-item wide">Item 2</div>
            <div class="flex-item wide">Item 3</div>
            <div class="flex-item wide">Item 4</div>
          </div>
        </div>
      </div>
      <div class="col-6">
        <div class="grid-item">
          <h5>No Wrap</h5>
          <p>Items stay on a single line and may overflow.</p>
          <div class="flex-demo flex-nowrap">
            <div class="flex-item wide">Item 1</div>
            <div class="flex-item wide">Item 2</div>
            <div class="flex-item wide">Item 3</div>
            <div class="flex-item wide">Item 4</div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="row">
    <div class="col-12">
      <div class="grid-item">
        <h5>Wrap Reverse</h5>
        <p>Items wrap to the previous line in reverse order.</p>
        <div class="flex-demo flex-wrap-reverse">
          <div class="flex-item wide">Item 1</div>
          <div class="flex-item wide">Item 2</div>
          <div class="flex-item wide">Item 3</div>
          <div class="flex-item wide">Item 4</div>
          <div class="flex-item wide">Item 5</div>
          <div class="flex-item wide">Item 6</div>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="demo-section">

```

<h4>Justify Content</h4>

<p>Control the alignment of flex items along the main axis.</p>

```
<div class="container">
  <div class="row">
    <div class="col-4">
      <div class="grid-item">
        <h5>Start (Default)</h5>
        <p>Items align to the start of the container.</p>
        <div class="flex-demo justify-content-start">
          <div class="flex-item">1</div>
          <div class="flex-item">2</div>
          <div class="flex-item">3</div>
        </div>
      </div>
    </div>
    <div class="col-4">
      <div class="grid-item">
        <h5>Center</h5>
        <p>Items align to the center of the container.</p>
        <div class="flex-demo justify-content-center">
          <div class="flex-item">1</div>
          <div class="flex-item">2</div>
          <div class="flex-item">3</div>
        </div>
      </div>
    </div>
    <div class="col-4">
      <div class="grid-item">
        <h5>End</h5>
        <p>Items align to the end of the container.</p>
        <div class="flex-demo justify-content-end">
          <div class="flex-item">1</div>
          <div class="flex-item">2</div>
          <div class="flex-item">3</div>
        </div>
      </div>
    </div>
  </div>

  <div class="row">
    <div class="col-6">
      <div class="grid-item">
        <h5>Between</h5>
        <p>Items are distributed with space between them.</p>
        <div class="flex-demo justify-content-between">
          <div class="flex-item">1</div>
          <div class="flex-item">2</div>
          <div class="flex-item">3</div>
        </div>
      </div>
    </div>
    <div class="col-6">
      <div class="grid-item">
        <h5>Around</h5>
        <p>Items are distributed with equal space around them.</p>
```

```

        <div class="flex-demo justify-content-around">
            <div class="flex-item">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item">3</div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

<div class="demo-section">
    <h4>Align Items</h4>
    <p>Control the alignment of flex items along the cross axis.</p>

    <div class="container">
        <div class="row">
            <div class="col-4">
                <div class="grid-item">
                    <h5>Start</h5>
                    <p>Items align to the start of the cross axis.</p>
                    <div class="flex-demo align-items-start">
                        <div class="flex-item tall">1</div>
                        <div class="flex-item">2</div>
                        <div class="flex-item tall">3</div>
                    </div>
                </div>
            </div>
            <div class="col-4">
                <div class="grid-item">
                    <h5>Center</h5>
                    <p>Items align to the center of the cross axis.</p>
                    <div class="flex-demo align-items-center">
                        <div class="flex-item tall">1</div>
                        <div class="flex-item">2</div>
                        <div class="flex-item tall">3</div>
                    </div>
                </div>
            </div>
            <div class="col-4">
                <div class="grid-item">
                    <h5>End</h5>
                    <p>Items align to the end of the cross axis.</p>
                    <div class="flex-demo align-items-end">
                        <div class="flex-item tall">1</div>
                        <div class="flex-item">2</div>
                        <div class="flex-item tall">3</div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-6">
            <div class="grid-item">
                <h5>Baseline</h5>

```

```

        <p>Items align to their baseline.</p>
        <div class="flex-demo align-items-baseline">
            <div class="flex-item baseline">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item baseline">3</div>
        </div>
    </div>
</div>
<div class="col-6">
    <div class="grid-item">
        <h5>Stretch (Default)</h5>
        <p>Items stretch to fill the container height.</p>
        <div class="flex-demo align-items-stretch">
            <div class="flex-item">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item">3</div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="demo-section">
    <h4>Align Self</h4>
    <p>Override the alignment for individual flex items.</p>

    <div class="container">
        <div class="row">
            <div class="col-12">
                <div class="grid-item">
                    <h5>Individual Item Alignment</h5>
                    <p>Each item can have its own alignment override.</p>
                    <div class="flex-demo">
                        <div class="flex-item tall align-self-start">Start</div>
                        <div class="flex-item align-self-center">Center</div>
                        <div class="flex-item tall align-self-end">End</div>
                        <div class="flex-item align-self-baseline">Baseline</div>
                        <div class="flex-item align-self-stretch">Stretch</div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
<div class="demo-section">
    <h4>Responsive Flexbox</h4>
    <p>Flexbox utilities with responsive breakpoint variants.</p>

    <div class="container">
        <div class="row">
            <div class="col-12">
                <div class="grid-item">
                    <h5>Responsive Flex Direction</h5>
                    <p>

```

```

        Direction changes based on screen size: column on mobile, row on
        larger screens.
    </p>
    <div class="flex-demo flex-column flex-md-row">
        <div class="flex-item">Mobile: Column</div>
        <div class="flex-item">Desktop: Row</div>
        <div class="flex-item">Responsive!</div>
    </div>
</div>
</div>
</div>

<div class="row">
    <div class="col-12">
        <div class="grid-item">
            <h5>Responsive Justify Content</h5>
            <p>
                Alignment changes based on screen size: center on mobile, between
                on larger screens.
            </p>
            <div
                class="flex-demo justify-content-center justify-content-md-between"
            >
                <div class="flex-item">Centered on mobile</div>
                <div class="flex-item">Between on desktop</div>
                <div class="flex-item">Responsive alignment!</div>
            </div>
        </div>
    </div>
</div>
</div>
</div>

<div class="demo-info">
    <h4>Flexbox Utility Features</h4>
    <ul>
        <li>
            <strong>Direction Control:</strong> Row, column, and reverse variants
        </li>
        <li><strong>Wrap Options:</strong> Wrap, nowrap, and wrap-reverse</li>
        <li>
            <strong>Justify Content:</strong> Start, end, center, between, around
        </li>
        <li>
            <strong>Align Items:</strong> Start, end, center, baseline, stretch
        </li>
        <li><strong>Align Self:</strong> Individual item alignment override</li>
        <li>
            <strong>Responsive Variants:</strong> All utilities available at each
            breakpoint
        </li>
    </ul>

    <h4>Common Use Cases</h4>
    <ul>
        <li>

```

```

        <strong>Navigation:</strong> Use justify-content-between for logo and
        menu
    </li>
    <li>
        <strong>Centered Content:</strong> Use justify-content-center and
        align-items-center
    </li>
    <li>
        <strong>Card Layouts:</strong> Use flex-wrap for responsive card grids
    </li>
    <li>
        <strong>Form Layouts:</strong> Use flex-column for stacked form fields
    </li>
    <li>
        <strong>Responsive Design:</strong> Combine with breakpoint classes for
        adaptive layouts
    </li>
</ul>
</div>
</div>

```

```

import { Component } from "@angular/core";

@Component({
  selector: "app-grid-flexbox",
  standalone: true,
  template: `
    <div class="grid-flexbox-demo">
      <h3>Flexbox Utilities</h3>

      <div class="demo-section">
        <h4>Flex Direction</h4>
        <p>Control the direction of flex items using direction classes.</p>

        <div class="container">
          <div class="row">
            <div class="col-6">
              <div class="grid-item">
                <h5>Row Direction (Default)</h5>
                <p>Items flow horizontally from left to right.</p>
                <div class="flex-demo flex-row">
                  <div class="flex-item">1</div>
                  <div class="flex-item">2</div>
                  <div class="flex-item">3</div>
                </div>
              </div>
            </div>
            <div class="col-6">
              <div class="grid-item">
                <h5>Column Direction</h5>
                <p>Items flow vertically from top to bottom.</p>
                <div class="flex-demo flex-column">
                  <div class="flex-item">1</div>

```

```

        <div class="flex-item">2</div>
        <div class="flex-item">3</div>
    </div>
</div>
</div>
</div>
</div>

<div class="row">
    <div class="col-6">
        <div class="grid-item">
            <h5>Row Reverse</h5>
            <p>Items flow horizontally from right to left.</p>
            <div class="flex-demo flex-row-reverse">
                <div class="flex-item">1</div>
                <div class="flex-item">2</div>
                <div class="flex-item">3</div>
            </div>
        </div>
    </div>
</div>
<div class="col-6">
    <div class="grid-item">
        <h5>Column Reverse</h5>
        <p>Items flow vertically from bottom to top.</p>
        <div class="flex-demo flex-column-reverse">
            <div class="flex-item">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item">3</div>
        </div>
    </div>
</div>
</div>
</div>
</div>

<div class="demo-section">
    <h4>Flex Wrap</h4>
    <p>Control how flex items wrap when they exceed the container width.</p>

    <div class="container">
        <div class="row">
            <div class="col-6">
                <div class="grid-item">
                    <h5>Wrap (Default)</h5>
                    <p>Items wrap to the next line when needed.</p>
                    <div class="flex-demo flex-wrap">
                        <div class="flex-item wide">Item 1</div>
                        <div class="flex-item wide">Item 2</div>
                        <div class="flex-item wide">Item 3</div>
                        <div class="flex-item wide">Item 4</div>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-6">
            <div class="grid-item">
                <h5>No Wrap</h5>
                <p>Items stay on a single line and may overflow.</p>

```



```

        <div class="flex-demo flex-nowrap">
            <div class="flex-item wide">Item 1</div>
            <div class="flex-item wide">Item 2</div>
            <div class="flex-item wide">Item 3</div>
            <div class="flex-item wide">Item 4</div>
        </div>
    </div>
</div>
</div>

<div class="row">
    <div class="col-12">
        <div class="grid-item">
            <h5>Wrap Reverse</h5>
            <p>Items wrap to the previous line in reverse order.</p>
            <div class="flex-demo flex-wrap-reverse">
                <div class="flex-item wide">Item 1</div>
                <div class="flex-item wide">Item 2</div>
                <div class="flex-item wide">Item 3</div>
                <div class="flex-item wide">Item 4</div>
                <div class="flex-item wide">Item 5</div>
                <div class="flex-item wide">Item 6</div>
            </div>
        </div>
    </div>
</div>
</div>
</div>

<div class="demo-section">
    <h4>Justify Content</h4>
    <p>Control the alignment of flex items along the main axis.</p>

    <div class="container">
        <div class="row">
            <div class="col-4">
                <div class="grid-item">
                    <h5>Start (Default)</h5>
                    <p>Items align to the start of the container.</p>
                    <div class="flex-demo justify-content-start">
                        <div class="flex-item">1</div>
                        <div class="flex-item">2</div>
                        <div class="flex-item">3</div>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-4">
            <div class="grid-item">
                <h5>Center</h5>
                <p>Items align to the center of the container.</p>
                <div class="flex-demo justify-content-center">
                    <div class="flex-item">1</div>
                    <div class="flex-item">2</div>
                    <div class="flex-item">3</div>
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
    <div class="col-4">
      <div class="grid-item">
        <h5>End</h5>
        <p>Items align to the end of the container.</p>
        <div class="flex-demo justify-content-end">
          <div class="flex-item">1</div>
          <div class="flex-item">2</div>
          <div class="flex-item">3</div>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="row">
  <div class="col-6">
    <div class="grid-item">
      <h5>Between</h5>
      <p>Items are distributed with space between them.</p>
      <div class="flex-demo justify-content-between">
        <div class="flex-item">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item">3</div>
      </div>
    </div>
  </div>
  <div class="col-6">
    <div class="grid-item">
      <h5>Around</h5>
      <p>Items are distributed with equal space around them.</p>
      <div class="flex-demo justify-content-around">
        <div class="flex-item">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item">3</div>
      </div>
    </div>
  </div>
</div>
</div>

<div class="demo-section">
  <h4>Align Items</h4>
  <p>Control the alignment of flex items along the cross axis.</p>

  <div class="container">
    <div class="row">
      <div class="col-4">
        <div class="grid-item">
          <h5>Start</h5>
          <p>Items align to the start of the cross axis.</p>
          <div class="flex-demo align-items-start">
            <div class="flex-item tall">1</div>
            <div class="flex-item">2</div>
            <div class="flex-item tall">3</div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    </div>
  </div>
  <div class="col-4">
    <div class="grid-item">
      <h5>Center</h5>
      <p>Items align to the center of the cross axis.</p>
      <div class="flex-demo align-items-center">
        <div class="flex-item tall">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item tall">3</div>
      </div>
    </div>
  </div>
  <div class="col-4">
    <div class="grid-item">
      <h5>End</h5>
      <p>Items align to the end of the cross axis.</p>
      <div class="flex-demo align-items-end">
        <div class="flex-item tall">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item tall">3</div>
      </div>
    </div>
  </div>
</div>

<div class="row">
  <div class="col-6">
    <div class="grid-item">
      <h5>Baseline</h5>
      <p>Items align to their baseline.</p>
      <div class="flex-demo align-items-baseline">
        <div class="flex-item baseline">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item baseline">3</div>
      </div>
    </div>
  </div>
  <div class="col-6">
    <div class="grid-item">
      <h5>Stretch (Default)</h5>
      <p>Items stretch to fill the container height.</p>
      <div class="flex-demo align-items-stretch">
        <div class="flex-item">1</div>
        <div class="flex-item">2</div>
        <div class="flex-item">3</div>
      </div>
    </div>
  </div>
</div>
</div>

<div class="demo-section">
  <h4>Align Self</h4>
  <p>Override the alignment for individual flex items.</p>

```

```

<div class="container">
  <div class="row">
    <div class="col-12">
      <div class="grid-item">
        <h5>Individual Item Alignment</h5>
        <p>Each item can have its own alignment override.</p>
        <div class="flex-demo">
          <div class="flex-item tall align-self-start">Start</div>
          <div class="flex-item align-self-center">Center</div>
          <div class="flex-item tall align-self-end">End</div>
          <div class="flex-item align-self-baseline">Baseline</div>
          <div class="flex-item align-self-stretch">Stretch</div>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="demo-section">
  <h4>Responsive Flexbox</h4>
  <p>Flexbox utilities with responsive breakpoint variants.</p>

  <div class="container">
    <div class="row">
      <div class="col-12">
        <div class="grid-item">
          <h5>Responsive Flex Direction</h5>
          <p>
            Direction changes based on screen size: column on mobile, row
            on larger screens.
          </p>
          <div class="flex-demo flex-column flex-md-row">
            <div class="flex-item">Mobile: Column</div>
            <div class="flex-item">Desktop: Row</div>
            <div class="flex-item">Responsive!</div>
          </div>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-12">
        <div class="grid-item">
          <h5>Responsive Justify Content</h5>
          <p>
            Alignment changes based on screen size: center on mobile,
            between on larger screens.
          </p>
          <div
            class="flex-demo justify-content-center justify-content-md-between"
            >
            <div class="flex-item">Centered on mobile</div>
            <div class="flex-item">Between on desktop</div>
            <div class="flex-item">Responsive alignment!</div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
    </div>
  </div>
</div>
</div>
</div>
```

```
<div class="demo-info">
  <h4>Flexbox Utility Features</h4>
  <ul>
    <li>
      <strong>Direction Control:</strong> Row, column, and reverse
      variants
    </li>
    <li><strong>Wrap Options:</strong> Wrap, nowrap, and wrap-reverse</li>
    <li>
      <strong>Justify Content:</strong> Start, end, center, between,
      around
    </li>
    <li>
      <strong>Align Items:</strong> Start, end, center, baseline, stretch
    </li>
    <li>
      <strong>Align Self:</strong> Individual item alignment override
    </li>
    <li>
      <strong>Responsive Variants:</strong> All utilities available at
      each breakpoint
    </li>
  </ul>
</div>
```

<h4>Common Use Cases</h4>

```
<ul>
  <li>
    <strong>Navigation:</strong> Use justify-content-between for logo
    and menu
  </li>
  <li>
    <strong>Centered Content:</strong> Use justify-content-center and
    align-items-center
  </li>
  <li>
    <strong>Card Layouts:</strong> Use flex-wrap for responsive card
    grids
  </li>
  <li>
    <strong>Form Layouts:</strong> Use flex-column for stacked form
    fields
  </li>
  <li>
    <strong>Responsive Design:</strong> Combine with breakpoint classes
    for adaptive layouts
  </li>
</ul>
```

```
</div>
```

```
</div>
```

```
,
```

```
styles: [
  \
    .grid-flexbox-demo {
      padding: 20px;
      max-width: 1200px;
      margin: 0 auto;
    }

    .grid-flexbox-demo h3 {
      margin: 0 0 20px 0;
      color: #333;
      font-size: 24px;
      font-weight: 600;
    }

    .demo-section {
      margin-bottom: 40px;
      padding: 20px;
      border: 1px solid #e5e7eb;
      border-radius: 8px;
      background: #f9fafb;
    }

    .demo-section h4 {
      margin: 0 0 12px 0;
      color: #333;
      font-size: 18px;
      font-weight: 600;
    }

    .demo-section p {
      margin: 0 0 20px 0;
      color: #555;
      line-height: 1.5;
    }

    .grid-item {
      background: #fff;
      border: 1px solid #e5e7eb;
      border-radius: 6px;
      padding: 20px;
      height: 100%;
      min-height: 150px;
      display: flex;
      flex-direction: column;
    }

    .grid-item h5 {
      margin: 0 0 12px 0;
      color: #333;
      font-size: 16px;
      font-weight: 600;
    }

    .grid-item p {
      margin: 0 0 15px 0;
```

```
    color: #666;
    line-height: 1.4;
}

.flex-demo {
  display: flex;
  background: #f8f9fa;
  border: 1px solid #dee2e6;
  border-radius: 4px;
  padding: 10px;
  min-height: 60px;
  gap: 10px;
}

.flex-item {
  background: #007bff;
  color: white;
  padding: 10px 15px;
  border-radius: 4px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-weight: 600;
  min-width: 40px;
  min-height: 40px;
}

.flex-item.wide {
  min-width: 80px;
}

.flex-item.tall {
  min-height: 60px;
}

.flex-item.baseline {
  font-size: 18px;
  padding-top: 20px;
}

.demo-info {
  margin-top: 40px;
  padding: 20px;
  background: #f8f9fa;
  border: 1px solid #dee2e6;
  border-radius: 8px;
}

.demo-info h4 {
  margin: 0 0 16px 0;
  color: #333;
  font-size: 18px;
  font-weight: 600;
}

.demo-info ul {
```

```

        margin: 0 0 20px 0;
        padding-left: 20px;
        color: #555;
    }

    .demo-info li {
        margin-bottom: 8px;
        line-height: 1.4;
    }

    .demo-info strong {
        color: #333;
    }
},
],
})
export class GridFlexboxComponent {
    constructor() {
        console.log("Grid Flexbox Component initialized");
    }
}

```

■ No code found

■ No code found